

## Market Price Prediction Based on Neural Network Using Hadoop MapReduce Technique

**Mukesh**

M Tech (CNE),

SJBIT

Bengaluru -560060

**Rohini T V**

Asst.Professor, Dept. Of ISE

SJBIT,

Bengaluru -560060

**Abstract:** Neural Networks (NN) has been used to solve wide variety of problems, one of them is stock market prediction. Neural Networks are modeled to predict future value of stock market using historical data. In this paper, a Multilayer perceptron (MLP) Neural Network is used to predict future stock market. Two MLP Neural Networks have been used, one is trained using Least Mean Square algorithm and another one using Sigmoid Delta algorithm. RMS error is calculated against actual and predicted value. Results have shown that MLP Neural Network with LMS training algorithm has lower RMS error than MLP Neural Network with sigmoid delta learning algorithm. The process of training large scale Neural Networks is both computation-intensive and data-intensive. Entire training workflow needs to carry out thousands of epochs' iteration, which makes it computationally expensive. On the other hand, in order to generate better results, large scale training datasets are usually used in applications. As a result, training large scale Neural Networks on a single PC is usually very time-consuming. To address this, MLP Neural Network model is implemented in MapReduce programming model with Hadoop framework. Since the MapReduce programming model has the ability to rapidly process large amount of data in parallel. Hence MapReduce works with Hadoop Distributed File System (HDFS). The entire process of predicting stock market using Neural Network in Hadoop's fully distributed mode and using HDFS the implementation results shows the effectiveness in reducing the time of prediction process with better accuracy.

**Keywords:** Neural network, hadoop, Multilayer perceptron, Least Mean Square, HDFS

### INTRODUCTION

Stock market prediction deals with determining the future value of a company or other financial instrument traded on a financial market. The successful prediction of a stock market future price could yield significant profit

An Artificial Neural Network (ANN) is a model composed of several highly interconnected computational units called neurons or nodes. Each node performs a simple operation on an input to generate an output that is forwarded to the next node in the sequence. This parallel processing allows for great advantages in data analysis. Artificial Neural Network are widely used in various branches of Engineering and Science and their property to approximate complex and nonlinear equations makes it a useful tools in econometric analysis.

Hadoop MapReduce is a latest framework specially designed for processing large datasets on distributed sources. Apache's Hadoop is an implementation of MapReduce. MapReduce is a programming model for expressing distributed calculation on massive amount of data and an execution framework for large-scale data processing on clusters of article of trade servers. It was originally developed by Google and built on well-known principles in parallel and distributed processing Hadoop is the open source implementation of MapReduce written in java which provides reliable, scalable and fault tolerance distributed computing. Hadoop environment set up involves a great number of parameters which are crucial to achieve excellent performance. It allows programmers to develop distributed applications without any knowledge. Key-value pair forms the basic data structure in MapReduce. Keys and values may be primitives such as integers, floating point values, strings, and raw bytes or they may be arbitrary complex structures (lists, tuples associative array, etc.)

### RELATED WORK

Training large scale neural networks on a single PC is usually most time consuming, which may take several days to weeks to finish and sometimes even cannot be done. Thus, the slow training speed of large scale [8] neural networks has limited their use for processing many complex and valuable problems in practice. From the intuition, it is usually recognized that training over large scale samples leads better learning results than over small amount of samples. Thus, for those [9] neural network-based applications, training large scale neural networks plays an important role in achieving optimal precisions and results

To Deal with large data set [5] Hadoop technique is used. It is a framework for running applications with large sized data on large clusters built of commodity hardware. The Hadoop framework transparently provides applications both reliability and data motion. Hadoop implements a computational paradigm named Map/Reduce. Map/Reduce process vast amounts of data (multi-terabyte data-sets) in parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The [13] MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. The applications specify the input/output locations and supply map and reduce functions via implementations

of appropriate interfaces and/or abstract-classes. These, and other job parameters, comprise the job configuration. Hadoop job client then submits the job (jar/executable etc.) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client. In addition, it provides a distributed file system (HDFS) that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both Map/Reduce and the distributed file system are designed so that node failures are automatically handled by the framework

### Methodology used

In this paper, goal is to build a Multilayer perceptron neural network model for stock market prediction and [8] Least Mean-Square (LMS) and Sigmoid Delta Algorithm for supervised learning are used and outcome is compared to decide which is best predictor

#### A. Feed forward neural network

It is an [9] artificial neural network where connections between the units do not form a directed cycle. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal, each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes. In this network, the information moves in only one direction, forward, from the input nodes. Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs.

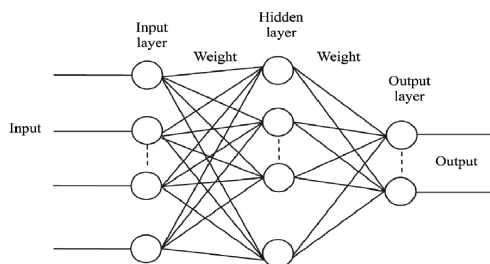


Figure 1 Feed forward neural network architecture

1. **Input Layer** The input layer to the neural network is the channel through which the external environment presents a pattern to the neural network. Every input neuron should represent some independent variable that has an influence over the output of the neural network.
2. **Output Layer** The output layer of the neural network is what actually presents a pattern to the external environment. Whatever pattern is presented by the output layer can be directly traced back to the input layer.
3. **Hidden Layer** Hidden layers can be simply inserted between the input and output layers. The hidden layers can also take on more complex structures. The only purpose of the hidden layers is to allow the neural

#### B. Multilayer perceptron

It is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A [9] MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique for training the network. Here, the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, it is said that the network has learned a certain target function. To adjust weights properly, it applies a general method for non-linear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated, and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function)

### PROPOSED SYSTEM

The proposed system contains two major technologies, first is the Artificial Neural Network which is a best predictor but it takes lot of iterations to yield better results, second is the Hadoop MapReduce programming which processes huge amount of data in parallel, executing large number of iterations in fraction of time. The proposed system uses Least Mean-Square and Sigmoid Delta Algorithm.

#### The proposed system works as follows:

1. The stock value series data is split into sequence of four values.
2. First 3 values are used as input and the fourth value is used as sample output.
3. Neural Network learning rule is set as LMS and neural network is trained.
4. Neural Network learning rule is set as Sigmoid and the neural network is trained.
5. For test data the stock market price is predicted with previous three day values.
6. The MSE for each neural network is calculated and plotted in a neat graph.

Details of the Least Mean-Square algorithm is as follows

#### Learning through Least Mean-Square [8](LMS) Algorithm:

LMS is an example of supervised learning algorithm in NN similar with the Perceptron learning algorithm. In the

perceptron learning algorithm, the algorithm trains the perceptron until it correctly classifies the output of the training set but LMS uses another termination criterion in order to train the perceptron. So instead of training the perceptron until a solution is found, another criterion is to continue training while the [8] Mean-Square Error (MSE) is greater than a certain value.

LMS is a fast algorithm that minimizes the MSE. The MSE is the average of the weighted sum of the error for N training sample which defined as:

$$MSE = \frac{\sum_{j=1}^N (R - C_j)^2}{N}$$

Where R is the output of the perceptron and Cj is the current test inputs.

In order to train the perceptron by using LMS, we can iterate the test set, taking a set of inputs, computing the output and then using the error to adjust the weight. This process can be done either randomly by the test set, or for each test of the set in succession. The learning rule of LMS is given as:

$$w_{t+1} = w + p(R - C)E$$

The learning rule adjusts the weight based on the error (R-C or expected output minus actual output). Once the error is calculated, the weights are adjusted by a small amount, p in the direction of the input, E. This has the effect of adjusting the weights to reduce the output error.

## IMPLEMENTATION

### MapReduce implementation of Neural Network:

The code in the Fig.2 is used to call neural network NN function for training and mapper contains code where NN function is called for prediction after NN returns predicted value, in mapper RMS error is calculated. MapReduce code it collects output from mapper and writes to a output file and in this mean value of both LMS and sigmoid delta algorithm error

Mapper class:

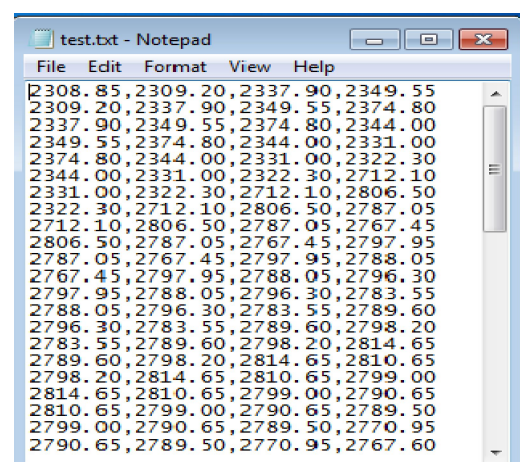
```
public void map(Object key, Text value, Context context)
throws IOException,
InterruptedException
{
String line = value.toString();
String[] parts = line.split(",");
Try
{
double[] price = new double[3];
price[0] = Double.parseDouble(parts[0]);
price[1] = Double.parseDouble(parts[1]);
price[2] = Double.parseDouble(parts[2]);
double actualPrice =
Double.parseDouble(parts[3]);
double predictedPrice =
PredictMarket.tnm.getPredictedPrice(price);
System.out.println("Predicted price in LMS is " +
predictedPrice);
```

```
double lmserror = Math.abs(actualPrice -
predictedPrice);
lmserror = Math.sqrt(lmserror);
predictedPrice = tsd.getPredictedPrice(price);
System.out.println("Predicted price in Sigmoid
delta is " + predictedPrice);
double sdeltaerror = Math.abs(actualPrice -
predictedPrice);
sdeltaerror = Math.sqrt(sdeltaerror);
context.write(word, new
DoubleWritable(sdeltaerror));
}
Reducer Class:
public void reduce(Text key, Iterable<DoubleWritable>
values, Context context) throws
IOException, InterruptedException
{
int count = 0; double sum = 0;
for (DoubleWritable doubleWritable : values)
{
double value = doubleWritable.get();
sum += value;
count++;
}
double average = sum / count;
context.write(key, result);
}
```

Fig. 2 Mapper And Reduce class of LMS

## RESULT ANALYSIS

In data preprocessing step, data is processed as shown in Fig. 3. Each row contains four values, first three values are given as input to neural network then neural network returns a predicted value. This value is subtracted with fourth value from preprocessed data and RMS error is calculated. In the next row of preprocessed data, first value from previous row is discarded and next four days value is considered it is done for entire stock values



2308.85	2309.20	2337.90	2349.55
2309.20	2337.90	2349.55	2374.80
2337.90	2349.55	2374.80	2344.00
2349.55	2374.80	2344.00	2331.00
2374.80	2344.00	2331.00	2322.30
2344.00	2331.00	2322.30	2712.10
2331.00	2322.30	2712.10	2806.50
2322.30	2712.10	2806.50	2787.05
2712.10	2806.50	2787.05	2767.45
2806.50	2787.05	2767.45	2797.95
2787.05	2767.45	2797.95	2788.05
2767.45	2797.95	2788.05	2796.30
2797.95	2788.05	2796.30	2783.55
2788.05	2796.30	2783.55	2789.60
2796.30	2783.55	2789.60	2798.20
2783.55	2789.60	2798.20	2814.65
2789.60	2798.20	2814.65	2810.65
2798.20	2814.65	2810.65	2799.00
2814.65	2810.65	2799.00	2790.65
2810.65	2799.00	2790.65	2789.50
2799.00	2790.65	2789.50	2770.95
2790.65	2789.50	2770.95	2767.60

Fig 3 Snapshot of Input data

### A. Neural Network Training

In this step neural network is trained with stock market data. In this work supervised learning method is used, this



contains set of input and desired output. The training sample is shown in table 1. MLP neural network consists of three inputs,nine hidden and one output layer. So first three values indicate inputs and fourth one is desired output, this training is done to both LMS and sigmoid delta learning algorithm

Input1	Input2	Input3	Expected
0.231	0.231	0.234	0.235
0.231	0.234	0.237	0.233
0.231	0.235	0.233	0.281
0.231	0.237	0.271	0.280
0.231	0.234	0.279	0.278
0.231	0.233	0.280	0.281
0.232	0.233	0.280	0.281
0.232	0.234	0.279	0.281

**Table 1 Training sample table**

In this step actual prediction is done with the test data. Two Multilayer perceptron network are used one network uses LMS learning algorithm and another one sigmoid delta algorithm, both are tested against test data to interpret which one gives best result. Output is shown below

Map processing the search input 2308.85, 2309.20, 2337.90

Input	[0.23088499999999998, 0.23092, 0.23379]
Output	0.2512623975861707
Predicted price in LMS	is 2512.623975861707
Input:	[2.3088499999999998E-5, 2.3092E-5, 2.3379E-5]
Output	0.12783948041345292
predicted price in Sigmoiddelta	1278.394804134529

**Table 2 shows input,output and predicted price**

#### A. Root Mean Square (RMS) Error:

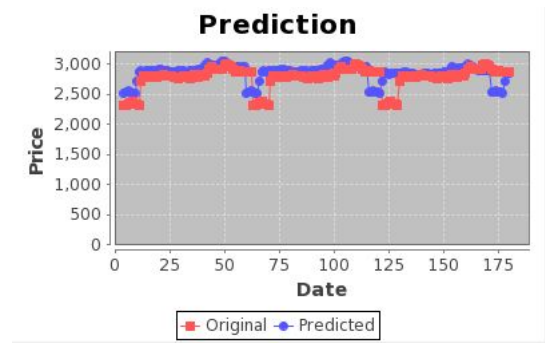
In this mean value of the RMS error is calculated. First RMS error is calculated using actual value (fourth value) in the test data and predicted value, this is calculated for every row in test data. It is done for both LMS and Sigmoid delta, later mean value of error for both methods is calculated, it is shown below. From table 3 it shows that LMS is better predictor than Sigmoid Delta method.

**Table 3 Shows error in prediction**

1	Sigmoid error	Delta	38.4409730775832
2	LMS error		6.77781808931945

#### B. Result Comparison

Graph is drawn by taking the actual stock market value and values predicted using LMS method. The graph shows both the values are similar. Graph is shown in Fig3.



**Fig 3 Comparison of Actual price versus predicted price results**

#### CONCLUSION AND FUTURE WORK

Stock market prediction is the act of trying to determine the future market price of a company stock. Artificial neural networks (ANN) are used to predict stock market. This work is mainly aimed to model a neural network to predict stock market, the neural network takes input as previous three days stock value and predicts the price for next day and the process is repeated. For this purpose a well-known two Multilayer Perceptron (MLP) neural network that has three inputs, nine hidden and one output layer is used. MLP is trained using two supervised learning algorithms, one is Least Mean Square (LMS) and second is Sigmoid Delta. RMS error is calculated for actual and predicted values. From the implementation result it shows that LMS has lower RMS error than sigmoid delta, So we can conclude that MLP neural network with LMS learning algorithm is best predictor.

On the other hand, in order to generate solid results, large scale training datasets are usually used in applications. As a result, training large scale neural networks on a single PC is usually very time-consuming, To address these challenges, new framework called Hadoop is used in this work. Hadoop is an open source software framework for storage and large scale processing of data. Its been observed that MLP neural network is implemented in Hadoop's Map/Reduce program that runs on multi node takes lesser CPU time than on single node hence prediction can be done efficiently on Hadoop.

The proposed work further can be extended by using different learning algorithms like Backpropagation, Perceptron Learning for training MLP Neural Network and their performance can be tested. This Neural Network is implemented on Hadoop framework, it's been observed that in multi node i.e. with three nodes, total CPU time taken for entire task of stock market prediction is less than running on single node, so by increasing the number of nodes CPU time efficiency can be improved. This work can be extended to predict weather, user web Behaviour Prediction and in the field of Medical science etc.

## REFERENCES

- [1]. Kunwar Singh Vaisla ,Dr. Ashutosh Kumar Bhatt, “An Analysis of the Performance of Artificial Neural Network Technique for Stock Market Forecasting”, International Journal on Computer Science and Engineering Vol. 02, No. 06, 2010, 2104-2109
- [2]. Kushagra Sahu, Revati Pawar,Stock Exchange IForecasting using Hadoop MAPREDUCE Technique, International Journal of Advancements in Research & Technology, Volume 2, Issue4, April-2013
- [3]. Puspanjali Mohapatra,Alok Raj “ Indian Stock Market Prediction Using Differential Evolutionary Neural Network Model”, International Journal of Electronics Communication and Computer Technology (IJECCCT) Volume 2 Issue 4 (July 2012)
- [4]. Rong Gu, Fura Shen, “A Parallel Computing Platform for Training Large Scale Neural Networks” 2013 IEEE International Conference on Big Data
- [5]. Dhruba Borthakur, The Hadoop Distributed File System:Architecture and Design
- [6]. Zhou Binhan, Wang Wenjun, Training Backpropagation Neural Network in MapReduce
- [7]. International Conference on Computer, Communications and Information Technology (CCIT) 2014
- [8]. Learning Algorithm for neural network Least Mean Square (LMS)  
<http://urrg.eng.usm.my>
- [9]. Artificial Neural Networks Technology, Training an Artificial Neural Networks  
<http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural4.html>
- [10]. Dhruba Borthakur, The Hadoop Distributed File System:Architecture and Design
- [11]. Dase R.K Application of Artificial Neural Network for stock market predictions International Journal of Machine Intelligence, ISSN: 0975-2927, Volume 2, Issue 2, 2010,pp-14-17
- [12]. Girish kumar Jha “Artificial Neural Network and its Applications
- [13]. [www.apache-hadoop.com](http://www.apache-hadoop.com)